

## WAX : A High Performance Spatial Auto-Correlation Application

Serge Monkewitz and Sherry Wheelock

*Infrared Processing and Analysis Center, California Institute of  
Technology, Mail Code 100-22, CA 91125*

**Abstract.** We describe the algorithms employed by WAX, a spatial auto-correlation application written in C and C++ which allows for both rapid grouping of multi-epoch apparitions as well as customizable statistical analysis of generated groups. The grouping algorithm, dubbed the swiss cheese algorithm, is designed to handle diverse input databases ranging from the 2MASS working point source database (an all sky database with relatively little coverage depth) to the 2MASS working calibration source database (a database with sparse but very deep coverage). WAX retrieves apparitions and stores groups directly from and to a DBMS, generating optimized C structures and ESQL/C code based on user defined retrieval and output columns. Furthermore, WAX allows generated groups to be spatially indexed via the HTM scheme and provides fast coverage queries for points and small circular areas on the sky. Finally, WAX operates on a declination based sky subdivision, allowing multiple instances to be run simultaneously and independently, further speeding the process of merging apparitions from very large databases. The Two Micron All Sky Survey will use WAX to create merged apparition catalogs from their working point and calibration source databases, linking generated groups to sources in the already publicly available all-sky catalogs. For a given 2MASS source, this will allow astronomers to examine the properties of many related (and as yet unpublished) 2MASS extractions, and further extends the scientific value of the 2MASS data sets.

### 1. Overview

The publicly released Two Micron All Sky Survey<sup>1</sup> (2MASS) catalogs, accessible via the NASA/IPAC Infrared Science Archive<sup>2</sup> (IRSA) website, were generated by running source extraction software on image data from many roughly rectangular regions on the sky. Many of these regions (scans) overlap each-other, meaning that a significant number of 2MASS sources were observed more than once (sometimes hundreds or thousands of times). In the 2MASS catalogs, multiple extractions (apparitions) corresponding to a single source were resolved by picking a *best*<sup>3</sup> apparition for catalog membership. Consequently, data for many

---

<sup>1</sup><http://www.ipac.caltech.edu/2mass/overview/access.html>

<sup>2</sup><http://irsa.ipac.caltech.edu/>

<sup>3</sup>[http://www.ipac.caltech.edu/2mass/releases/allsky/doc/sec5\\_4.html](http://www.ipac.caltech.edu/2mass/releases/allsky/doc/sec5_4.html)

2MASS extractions are unavailable and astronomers cannot examine the set of extractions corresponding to each source.

The Working Auto-Correlation<sup>4</sup> (WAX) software addresses these issues by generating groups of apparitions likely to correspond to distinct astronomical sources. Groups are made available as a catalog in conjunction with the database of extracted apparitions. Furthermore, the grouping algorithm employed is conservative: if an apparition is assigned to more than one group, then the apparition and its containing groups are flagged as confused. Attempts at resolving confusion are deferred so as not to impose a particular algorithm on the astronomer.

## 2. Architecture

WAX is a portable C/C++ application which performs I/O directly to and from a relational database management system (RDBMS). As of this writing, support is limited to the Informix RDBMS. By avoiding intermediate representations of input and output data, disk space requirements and time spent on I/O are both drastically reduced. Another benefit is that output can be queried and served immediately, without requiring the intervention of a database administrator.

The grouping algorithm (dubbed the swiss cheese algorithm) is hardwired, and places minimal constraints on the input RDBMS table: a per-apparition unique identifier and position suffice to run WAX. The output consists of between two and four tables :

**Group Catalog:** A table containing a unique identifier, apparition count, and confusion flag for each group. Other attributes (such as an average position) may also be computed by a user supplied plug-in.

**Link Catalog:** A table containing group/apparition identifier pairs. This table allows identifiers for all the apparitions belonging to a particular group to be retrieved. Similarly, identifiers for all groups containing a particular apparition can be retrieved.

**Singleton Catalog:** An optional table containing groups with just a single member apparition.

**Grouped Apparition Catalog:** An optional table mapping each multiply-assigned apparition to a *preferred* group (as determined by a user specified plug-in).

WAX itself does not perform computation of group attributes or choose *preferred* groups for apparitions. Instead, this task is left to a plug-in. At compile-time, the plug-in provides a retrieval and output table column specification. This specification is used to generate C structures corresponding to rows in the input and output tables. The specification is also used to generate high performance database I/O code (ESQL/C). At run-time, groups and apparitions are passed

---

<sup>4</sup><http://irsa.ipac.caltech.edu/applications/2MASS/WAX/docs/html/>

to the plug-in via the generated data structures, allowing for computation of arbitrary group attributes (e.g. average position, mean magnitudes, etc.). Because the complexity of I/O and data representation is hidden from the plug-in, its software interface is small and simple: the entire interface is specified with just five C function prototypes. To further simplify plug-in implementation, libraries are provided for common tasks such as computing the observational coverage and spatial index of a position.

Taken together, these features allow the WAX application to be tailored to specific apparition databases. Although initially developed to generate 2MASS data products, WAX is general enough to support other missions, and is expected to be employed by the Wide-field Infrared Survey Explorer<sup>5</sup> (WISE) team.

### 3. The Swiss Cheese Algorithm

The swiss cheese algorithm first computes both a *density* and a *centroid* for each apparition on the sky. The *density* for an apparition  $a$  is defined as the number of apparitions within a user specified angular distance  $\theta$  of  $a$ . The *centroid* is defined as the average position of the apparitions contributing to the *density* of  $a$ .

Next, the apparitions are sorted into decreasing *density* order. Each apparition in the resulting queue is said to be a *seed*; that is, an unprocessed apparition from which a group may be generated. A *group* is formally defined as the set of apparitions within a user specified distance  $\phi$  of a *seed centroid*.

*Groups* are constructed as follows :

1. Generate a *group*  $g$  from the *seed*  $s$  at the head of the queue.
2. Apparitions assigned to  $g$  are removed from the queue of *seeds*. Note that  $s$  (the head of the queue) will always be removed.
3. While the queue is non-empty, repeat steps 1 and 2.

In some circumstances, the algorithm assigns a given apparition to more than one group. Apparitions belonging to two or more groups (as well as groups containing such apparitions) are said to be *confused*. Figure 1 illustrates the swiss cheese algorithm and show how *confusion* arises.

Since the apparition databases being processed are far too large to fit in memory, they must be partitioned in some way. The WAX software splits the sky into declination bands to reduce the working set of apparitions, and, instead of processing *seeds* strictly in density order, works in spatial order while maintaining *density* order results. This allows for efficient computation of *densities*, *centroids*, and *groups*, each of which involve finding apparitions within some small radius of a position. Furthermore, it allows individual declination bands to be processed in parallel (non-adjacent bands are guaranteed to be data-independent). Figure 2 depicts the spatial subdivision scheme and traversal order used by WAX.

*Groups* are generated and *seeds* are discarded if and only if doing so does not violate the density ordering constraints imposed by the swiss cheese algorithm :

1. A *group* is generated around a *seed centroid* if and only if the *seed* cannot be a member of any *group* generated from a *denser seed*.

---

<sup>5</sup><http://www.astro.ucla.edu/~wright/WISE/>

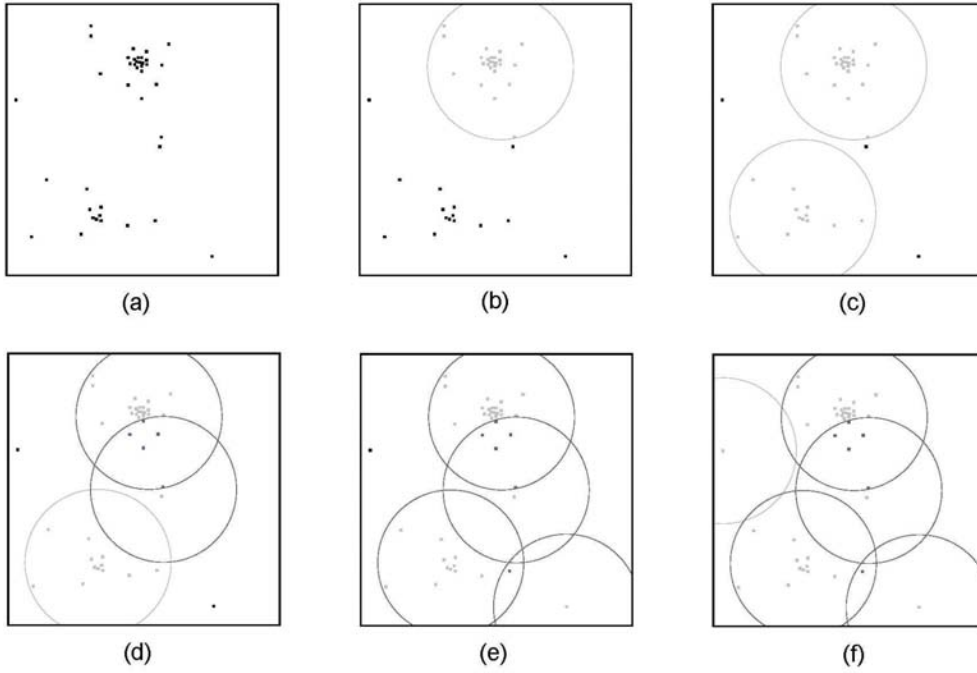


Figure 1. The swiss cheese algorithm operating on a small region of the sky. (a) The unprocessed sky. (b-c) The first and second *groups* are generated. Unprocessed apparitions are drawn in black, unconfused apparitions and *groups* in light grey. (d-f) As more *groups* are generated, confusion - drawn in dark grey - appears.

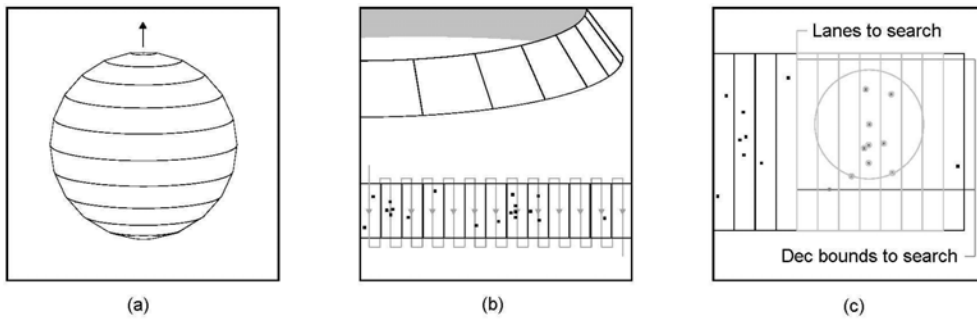


Figure 2. (a) The sky is subdivided into declination bands. (b) Each band is subdivided into lanes which are traversed in spatial order, indicated by the light grey line. (c) Only a small number of apparitions are considered to find a *density*, *centroid*, or *group*.

2. When a *group* is generated around a *seed*  $s$ , then an apparition assigned to  $s$  is discarded from the set of *seeds* if and only if it is less *dense* than  $s$ .

Because the sky cannot be considered in its entirety, there are cases where processing *seeds* in spatial order will produce different results than *density* order. In such cases, WAX attempts to minimize the deviation incurred and also flags each *group* generated out of order. Note that such cases did not occur during 2MASS data processing, and have yet to be encountered in practice.

#### 4. Performance

The WAX software has been used to generate group catalogs for the 2MASS Point, Extended, 6X, and Calibration Working Source Databases. In particular, a pre-filtered copy of the 2MASS Working Point Source Database containing roughly 800 million apparitions was processed by 4 instances of WAX running in parallel on a single 4 CPU machine in approximately 4.5 days.